EL049224012US Docket No. CISCO-4509

# UNITED STATES PATENT APPLICATION

FOR

# RESOLVING LBIST TIMING VIOLATIONS


**INVENTORS:**

**Xinli Gu, a citizen of China**
**Sung Soo Chung, a citizen of the United States of America**
**Frank Tsang, a citizen of the United States of America**


**ASSIGNED TO:**

**Cisco Technology, Inc., a California Corporation**


PREPARED BY:

**THELEN, REID, & PRIEST, LLP**
**P.O. BOX 640640**
**SAN JOSE, CA 95164-0640**
**TELEPHONE: (408) 292-5800**
**FAX: (408) 287-8040**


**Attorney Docket Number: CISCO-4509**

**Client Docket Number: CISCO-4509**

# SPECIFICATION

## RESOLVING LBIST TIMING VIOLATIONS

### FIELD OF THE INVENTION

**[0001]** The present invention is generally directed to the testing of integrated circuits. More particularly, the present invention is directed to the testing of integrated circuits using logic built-in self test.

### BACKGROUND OF THE INVENTION

**[0002]** Testing of integrated circuits has long been known in the art. Presently, testing techniques are in widespread use for nearly all types of integrated circuits ranging from those implemented in household toys and appliances to sophisticated supercomputers. As a result, the testing of integrated circuits forms an important part of the manufacturing process for most integrated circuits in use today.

**[0003]** These testing techniques have been developed in a wide variety of electronic manufacturing and circuit design configurations, depending upon the intended need at the implementation site. One form of testing of integrated circuits in use today is known as Logic Built-In Self Test (LBIST). Generally, LBIST is a technique that is used to test an integrated circuit such as an Application Specific Integrated Circuit (ASIC) by a test logic sub-circuit that is built into an overall circuit design. In LBIST, test vectors are generated by a Pseudo-Random Pattern Generator (PRPG), such as the LBIST Architect tool from Mentor Graphics of Wilsonville, Oregon, on an integrated circuit and the responses from the ASIC are compressed by a Multiple Input Shift Register (MISR) to form a "signature". The signature in the MISR will then be compared with a predetermined signature generally referred to as a "golden signature" to determine if there are any defects in the ASIC.

**[0004]** This technique has the benefits of: 1) improving test quality by performing at-speed testing; 2) reducing manufacturing test costs by reducing tester usage time, tester memory requirements and the required number of tester input/output connections; and 3) reducing system test, diagnosis and repair costs by performing ASIC testing on board the ASIC. The technique,

however, also has several disadvantages, including: 1) design hardware overhead due to the test logic inserted into the ASIC; and 2) extra design efforts due to the design complexity introduced. While the design hardware overhead is often negligible compared to design sizes and the silicon technologies used today, the extra design effort poses a significant obstacle to the implementation and usage of LBIST techniques.

**[0005]** Generally, one of the most significant obstacles in LBIST design is timing closure, arising from timing violations introduced by inserting the LBIST circuitry into the ASIC. The main timing violations introduced by LBIST are due to: 1) test point insertions; 2) x-bounding logic insertions; and 3) multi-cycle paths.

**[0006]** The test point insertion timing violations result when test points are inserted into circuit paths, such as for control and observation purposes. Due to the use of the pseudo-random pattern generator, random resistant logic in a design can be very difficult to test, making high test coverage technically and economically impractical. As stated above, these test point insertions can introduce functional path timing violations.

**[0007]** The second type of timing violation in LBIST design is caused by x-bounding logic. The nature of signature compression using a MISR requires that no unknown "X" values should be propagated into the MISR, otherwise the signature will be corrupted. Generally, "X" values are introduced by floating nets in a design, latches, non scannable flip-flops, external inputs which are not controlled on a tester (such as Credence QUARTET and Credence DUO available from Credence Systems Corporation of Fremont, California, and Teradyne J750 and Teradyne Catalyst 400 available from Teradyne, Inc. of Boston, Massachusetts) or system board, memories which are treated as black boxes and buses where contentions could happen. The x-bounding logic is used to block the "X" value propagation, however, the x-bounding logic could adversely affect design timing and, in many designs, timing budgets for memory accesses and I/O are often very tight.

**[0008]** Another type of timing violation in LBIST design is related to design false paths and multi-cycle paths. In functional mode, the switch of timing analysis for these paths is turned off. In other words, even though there are timing violations in these paths, they are ignored during the

functional mode timing analysis. In test mode, however, all paths in the design need to be tested, and therefore their timing should be considered. In normal scan design, this is generally not a significant problem because the test clock frequencies are very slow. In LBIST design however, the test clock frequencies are at or close to the functional clock frequencies of the integrated circuit and can therefor cause timing violations.

**[0009]** A practical issue in LBIST design is the fault coverage that can be achieved. Fault coverage is often limited due to the pseudo-random test vectors used. Currently, several approaches have been used to alleviate this situation. One approach is to insert more test points to achieve full testability. As described above, test point insertion can affect design timing. Another approach is to increase the number of pseudo-random vectors to be used. This approach, however is limited to situations when the tester cost is not an issue. In many cases, however, tester cost is of significant concern. Thus, using current techniques, a trade-off must be made in order to achieve optimal results with limited resources.

**[00010]** Presently, the foregoing practical issues have not been fully addressed in the existing art. One current approach is to minimize the number of test points to be inserted in order to reduce the number of timing violations caused by test points. A disadvantage of this approach is that there is no guarantee that all of the timing violations caused by test points will be resolved.

**[00011]** Another existing approach addresses the x-bounding timing violations by suppressing the clock to avoid capturing "X" values that will propagate into the MISR. Unfortunately, adding logic to clock trees to suppress capturing will increase clock tree skew. Yet another approach attempts to provide a solution to multi-cycle path timing problems, based on a list of multi-cycle paths and the information on the number of cycles for each multi-cycle path to disable the capturing of "X" values. Under this approach, by suppressing the capture of a transmitting flip-flop, the transmitting flip-flop holds its previous state for an extra number of cycles to allow the receiving flip-flop to capture data safely. When the transmitting flip-flop performs the capture, the capture of the receiving flip-flop will be suppressed so that there is no potential for a hold-time violation. Although this approach can achieve full coverage of a circuit, it is not without its shortcomings. First, the multiplexer added in the front of the transmitting flip-flops could cause its

own timing problems. Second, as a practical matter it is very hard to obtain the multi-cycle paths and their multi-cycle number information. In many cases, ASIC designers use a very simplified approach to specify the paths that can be ignored by for example setting all paths from a clock domain A to a clock domain B to false paths. Under this approach, however, it is very hard to obtain the multi-cycle paths and their multi-cycle number information because of the generally large number of specified paths involved. In addition, prior to manufacturing a circuit, circuit layout mapping processes, such as Place and Route processes, are generally used to determine the efficient positioning and connection paths of circuit components. These processes can also introduce additional timing violations that cause an undesirable signature mismatch between the signature in the MISR and the golden signature.

**[00012]** The present invention introduces a technique based on timing analyses to resolve timing problems efficiently so that extra design efforts introduced by using LBIST can be reduced or eliminated.

## SUMMARY OF THE INVENTION

**[00013]** A method and apparatus for resolving timing violations introduced by a logic built-in self test (LBIST) sub-circuit formed within an underlying integrated circuit includes analyzing a circuit path-list corresponding to the integrated circuit for timing violations and generating a circuit timing violations analysis output; generating a first LBIST/circuit path-list based on the circuit path-list and an LBIST path-list corresponding to the LBIST sub-circuit; analyzing the first LBIST/circuit path-list for timing violations and generating an LBIST/circuit timing violations analysis output; comparing the LBIST/circuit timing violations analysis output with the circuit timing violations analysis output; generating an LBIST/circuit constraint file based on the comparison and predetermined protocols; and generating a second LBIST/circuit path-list based on the circuit path-list, the LBIST path-list and the constraints file.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[00014]** The accompanying drawings, which are incorporated into and constitute a part of this specification, illustrate one or more exemplary embodiments of the invention, and together with the detailed description, serve to explain the principles and exemplary implementations of the invention.

**[00015]** In the drawings:

FIG. 1 is a prior art LBIST design flow diagram.

FIG. 2 illustrates an LBIST design flow diagram in accordance with one embodiment of the present invention.

FIG. 3 is a block circuit diagram of LBIST architecture diagram in accordance with one embodiment of the present invention.

FIG. 4 is a circuit diagram of logic inserted for control points diagram in accordance with one embodiment of the present invention.

FIG. 5 is a circuit diagram of logic inserted for observe points diagram in accordance with one embodiment of the present invention.

FIG. 6 is a circuit diagram for x-bounding logic diagram in accordance with one embodiment of the present invention.

FIG. 7 is a circuit diagram illustrating the suppression for blocking "X" propagation diagram in accordance with one embodiment of the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

**[00016]** Various exemplary embodiments of the present invention are described herein in the context of resolving timing violations introduced by LBIST. Those of ordinary skill in the art will realize that the following detailed description of the present invention is illustrative only and is not intended to be in any way limiting. Other embodiments of the present invention will readily suggest themselves to such skilled persons having the benefit of this disclosure. Reference will now be made in detail to exemplary implementations of the present invention as illustrated in the accompanying drawings. The same reference indicators will be used throughout the drawings and the following detailed descriptions to refer to the same or like parts.

**[00017]** In the interest of clarity, not all of the routine features of the exemplary implementations described herein are shown and described. It will of course, be appreciated that in the development of any such actual implementation, numerous implementation-specific decisions must be made in order to achieve the developer's specific goals, such as compliance with application- and business-related constraints, and that these specific goals will vary from one implementation to another and

from one developer to another. Moreover, it will be appreciated that such a development effort might be complex and time-consuming, but would nevertheless be a routine undertaking of engineering for those of ordinary skill in the art having the benefit of this disclosure.

**[00018]** Referring now more particularly to the Drawings, the present invention is directed to resolving timing violations introduced by a logic built-in self test (LBIST) sub-circuit formed within an underlying integrated circuit.

**[00019]** FIG. 1 illustrates a prior art LBIST design flow diagram, with the LBIST-related portions illustrated in shaded blocks. As shown, the flow begins at block 10 containing a conventional Register Transfer Logic (RTL) netlist, which is then forwarded to and received by block 12 where it undergoes conventional synthesis for generation of a gate level netlist in output block 14. Process then flows to the LBIST-related block 16 where a conventional scan & LBIST insertion is performed on the netlist to create the netlist with LBIST in output block 18. Next, at block 20, a conventional timing analysis is performed on the netlist with LBIST. This timing analysis generally includes two timing analyses: one in a functional mode and one in an LBIST mode. If the analysis in either mode shows a timing violation, then the process flow proceeds in a feedback path to block 12 at which point synthesis constraints are modified, thereafter the flow continues back to block 20 for further timing analysis as described above. If a timing violation still exists, the flow is returned back to block 12 at which point synthesis constraints are further modified. It may also be necessary for the flow to proceed in a feedback path to block 10 for modification to the RTL design. If modifications of either synthesis constraint or RTL design result in no timing violations at the timing analysis performed in block 20, then the flow proceeds to the output of a final netlist with LBIST in block 22. One shortcoming of this prior art approach is that often many iterations are needed to resolve the timing violations, resulting in expenditure of additional efforts, time-delays and other expenses associated with processing a number of iterations through the feedback flow and the subsequent modifications that are often required to resolve all timing violations for the netlist with LBIST in block 18. In addition, neither the modifying of synthesis constraints or the RTL design can guarantee the resolution of all of the timing violations. As explained below in greater detail, the present invention introduces a fast timing closure

technique to quickly resolve timing violations associated with LBIST, to reduce or eliminate the foregoing problems.

**[00020]** FIG. 2 illustrates a design flow diagram in accordance with one embodiment of the present invention, where the shaded portions are LBIST related. As before, an RTL 26 is processed for synthesis at block 28 resulting in a netlist at block 30. Once the gate level netlist at block 30 is ready, it is output to and received by block 42 where a primary timing analysis is performed on the netlist (such as by using the PrimeTime tool from Synopsys, Inc. of Mountain View, California), the results of which are output to block 44. The timing analysis results of block 44 will be later used to separate timing violations introduced by LBIST insertion from the violations that exist in the original netlist. In parallel with the primary timing analysis of block 42, the netlist is also passed to block 32 where scan and LBIST insertion are performed on the received netlist to generate a netlist with LBIST for output to block 34, with no LBIST constraints during this initial LBIST insertion. Next, in block 36 a secondary timing analysis is performed (such as by using the PrimeTime tool) on the received netlist with LBIST from block 34. The secondary timing analysis in block 36 generally includes two timing analyses: one in a functional mode and one in an LBIST mode, the results of which are then output to block 46 and block 48, respectively.

**[00021]** Next, in comparison block 50, the results of primary timing analysis from blocks 44 and secondary timing analysis 46 and 48 are compared to determine the existence of any timing violations. In an exemplary embodiment, the result of functional mode secondary timing analysis of block 46 is first compared with the primary timing analysis results of block 44 to determine the existence of a first predetermined set of timing violations such as control-point timing violations, observe-point timing violations and x-bounding timing violations. A second comparison of LBIST secondary timing analysis of block 48 is also compared with the functional mode timing analysis results of block 46 to determine the existence of a second predetermined set of timing violations such as multi-cycle timing violations and false-path timing violations.

**[00022]** If no timing violation is detected in either comparison performed in comparison block 50, then the flow proceeds to block 40 where a final netlist with LBIST is output. As explained below in greater detail, if a timing violation is detected in comparison block 50, then the results of

both comparisons are then used to generate an LBIST insertion constraint file of block 38. The LBIST insertion constraint file of block 38, in turn, is used to guide a second scan and LBIST insertion once the flow returns to block 32. A second netlist with LBIST is then generated and output to block 34. As explained below in greater detail, this second LBISTed netlist will result in no timing violations for both functional mode analysis and LBIST mode analysis, provided that the initial netlist is free of timing violations. Otherwise, the remaining timing violations are those that exist in the original netlist. Once another secondary timing analysis is performed in block 36 and no timing violations are detected, the flow then proceeds to block 40 where a final netlist with LBIST is output.

**[00023]** One advantage of the foregoing feature over the prior art is that it guarantees timing closure for LBIST insertion without any iteration between synthesis of block 28 and the LBIST portions of shaded blocks 32-38. In addition, it also guarantees that at most only one iteration in the illustrated flow between block 32 and block 38 is required to resolve all timing violations due to LBIST insertion. In an exemplary embodiment of the present invention, once the second netlist with LBIST in block 32 is generated, the flow proceeds directly to block 40 in which a final netlist with LBIST is output.

**[00024]** The various features and operations of the invention will now be described in greater detail in conjunction with the following figures.

**[00025]** FIG. 3 is a block diagram of a typical LBIST circuit architecture. In the exemplary embodiment shown, the LBIST architecture is based on Self-Test Using MISR/Parallel SRSG (shift register sequence generator) or "STUMPS" for short. As shown, the scan chains 58 of the core design 52 form STUMPS channels that are fed by either pseudo-random patterns from a pseudo-random pattern generator (PRPG) unit, or fed by a deterministic automatic test pattern generator (ATPG), such as the Fastscan tool available from Mentor Graphics, through scan_input pins in unit 54. The captured responses of the scan cells in the STUMPS channels are compressed in the MISR 56. To minimize or eliminate timing problems at the boundaries of the core design 52, lockup-latches 62 are inserted at the beginning and at the end of the STUMPS channels.

**[00026]** FIG. 4 illustrates an exemplary circuit diagram of logic inserted for control points. As shown control points can be of AND 64 or OR 66 type for control value “0” or control value “1”.

**[00027]** FIG. 5A and FIG. 5B illustrate an exemplary circuit diagram of logic inserted for observe points. In an exemplary embodiment as shown in FIG. 5A and FIG. 5B, two different types of logic are inserted for observation points to observe at flip-flops 74 and 72 respectively, where flip flop 74 is a dedicated flip-flop and flip-flop 72 is a functionally shared flip-flop. In the configuration of FIG. 5B, the observe enable signal is received by the observe-en line at the input port 76 of the AND gate , along with the XOR-tree input.

**[00028]** FIG. 6 illustrates an exemplary circuit diagram for x-bounding logic using a multiplexer and an existing scan cell. Another important requirement for a successful design tested by using LBIST is that there can be no propagation of “X” values to scan cells in the STUMPS chains as shown in FIG. 3. If an “X” value is captured into a scan cell, it will corrupt the MISR signature. Therefore, in analyzing the core design, x-bounding logic is inserted to block any x-sources by gating off internal x-sources 80 via multiplexing in a new signal from an existing scan cell. Primary inputs are also considered x-sources, and they are always x-bounded by multiplexing in data from existing scan cells. One advantage of using the multiplexing technique is increased controllability. In an exemplary embodiment, the multiplexing technique is used for all internal x-sources as well.

**[00029]** Returning to FIG. 2, the details of the generation of the LBIST constraint file in block 38 will now be described in greater detail. As disclosed above, prior to reaching block 38, three timing analyses are carried out. The first timing analysis performed in block 42 (the primary timing analysis) is performed for the netlist before LBIST insertion. The second and third timing analyses are those performed in block 36 for the netlist after LBIST insertion, with one analysis performed in the functional mode and the other performed in the LBIST mode. This includes the mode change and timing analysis for paths including all false paths and multi-cycle paths. As explained above, the LBIST constraint file is obtained from the comparison of the three timing analysis reports performed in block 50.

**[00030]** Next, the specific locations in the integrated circuit responsible for causing the timing violations are identified in the LBIST constraint file 38. To better illustrate obtaining the LBIST constraint file 38 from the three timing analyses, and identifying the specific locations in the integrated circuit responsible for causing timing violations, a set of definitions have been provided below. The mathematical notations used therein and their attributed meaning are in conformity with well known mathematical notations used in the field.

**[00031]** Definition_1: A path P is a sequence of device instances $D_1$ , $D_2$ , ..., $D_n$ that are connected in the sequence; $P = D_1 , D_2 , ..., D_n$ where $D_1$ is a flip-flop, a memory output or external input, $D_n$ is a flip-flop, a memory input or external output, and others are combinational devices. The sequence order is denoted as $D_i$ -> $D_j$ if $D_i$ is prior to $D_j$ in the sequence.

**[00032]** Definition_2: A test point TP is a device D that can be inserted into a path for testability purpose. A test point can be either a control point, an observe point or an X-bounding.

**[00033]** Definition_3: A test point TP = D is on a path $P = D_1, D_2 , ..., D_n$ if and only if $\exists\ i \in [1{:}n]$ : $D = D_i$. denoted by $TP \in P$.

**[00034]** Definition_4: Given a path $P_1 = D_{11}, D_{12} , ..., D_{1n}$ , path $P_2 = D_{21}, D_{22} , ..., D_{2(n+1)}$ is path $P_1$ with a test point TP = D inserted if and only if $D_{1i} = D_{2i}$ , $i \in [1,a]$, $1 \leq a \leq n$, $D_2(a+1) = D$, and $D_{1j} = D_2(j+1)$ , $j \in [a+1,n]$. This is denoted by $P_1 \subset P_2$. Otherwise, $P_2$ is not the same path as $P_1$ with a test point inserted, and it is denoted by $P_1 \not\subset P_1$ Path $P_1$ can have several test points inserted into it $P_1, P_2 , ..., P_n$. This is also denoted by $P_1 \subset P_n$. This definition therefore determines whether two given paths are different from one another in a feature other than an inserted test point.

**[00035]** Definition_5: $P_{func}$ is a set of timing violation paths reported from timing analysis in functional mode for the netlist before LBIST insertion (i.e. netlist 30 of block 44).

**[00036]** Definition_6: $P_{lbist}$ is a set of timing violation paths reported from timing analysis in functional mode for the netlist after LBIST insertion (i.e. netlist 34 of block 46). Static test signals controlling LBIST operations are included in a false path list during timing analysis for the netlist after LBIST insertion. Therefore they should not appear in P lbist.

**[00037]** Definition_7: $P_{false}$ is a set of timing violation paths reported from timing analysis in LBIST mode after LBIST insertion (i.e. netlist 34 of block 48).

**[00038]** The LBIST constraint file 38 is generated in accordance with the following heuristics referencing the foregoing definitions :

Heuristic_1: $P_{lbist}$ - $P_{func}$ is a set of functional timing violation paths caused by LBIST insertion:

$$P_{lbist} - P_{func} = \{ P \mid (P \in P_{lbist}) \wedge (\forall P_i \in P_{func} : (P_i \not\subset P) \wedge (P_i \neq P) ) \}$$

$P_{lbist}$ - $P_{func}$ is thus a subset of $P_{lbist}$ which contains the paths that are not in $P_{func}$ or the paths in $P_{func}$ with test points inserted. Therefore they are new timing violation paths after LBIST insertion.

**[00039]** In an exemplary embodiment, the paths from P func with test points inserted are not included in $P_{lbist}$ - $P_{func}$, and their timing violations will not be considered from LBIST insertion point of view until functional timing violations are resolved. This however, could introduce an extra iteration of timing analysis. In addition, after resolving timing violations for the netlist of block 30, LBIST insertion in block 32 could cause another set of timing violations. One solution is to include the paths in P lbist - P func so that LBIST insertion will not insert test points into these paths, thus to make their timing violations only functional related. Thus

$$P_{lbist} - P_{func} = \{ P \mid (P \in P_{lbist}) \wedge (\forall Pi \in P_{func} : P\,i \neq P) \}$$

**[00040]** Converting the timing violation paths caused by LBIST insertion to the LBIST constraint file of block 38, begins by identifying which control points, observe points and x-boundings inserted have caused timing violations. Assuming that $TP_{ctrl}$ , $TP_{obs}$ and $TP_{xbnd}$ are a set of control points, observe points and x-boundings inserted in the initial LBIST insertion. following three heuristics are used to perform this conversion.

**[00041]** Heuristic_2: The set of control points that were initially inserted and cause timing violations can be identified by

$$TP_{ctrl}\_vio = \{ TP \mid (TP \in TP_{ctrl}) \wedge (\exists P \in ( P_{lbist} - P_{func} ): TP \in P ) \}$$

Heuristic_3: The set of observe points that were initially inserted and cause timing violations can be identified by

$$TP_{obs}\_vio = \{\ TP \mid (TP \in TP_{obs}) \wedge (\exists\ P \in (\ P_{lbist} - P_{func}):\ TP \in P\ )\ \}$$

Heuristic_4: The set of x-boundings that were initially inserted and cause timing violations can be identified by

$$TP_{xbnd}\_vio = \{\ TP \mid (TP \in TP_{xbnd}) \wedge (\exists\ P \in (\ P_{lbist} - P_{func}):\ TP \in P\ )\ \}$$

$TP_{ctrl}\_vio$ , $TP_{obs}\_vio$ and $TP_{xbnd}\_vio$ will be included in the LBIST insertion constraint file 38. The constrained LBIST insertion will use this information to ascertain that no timing violation is caused by the constrained LBIST insertion.

**[00042]** In an exemplary embodiment, timing violations in false paths or multi-cycle path, are all considered as multi-cycle path timing violations and can be identified by the following heuristic:

Heuristic_5: The set of timing violation multi-cycle paths in LBIST mode can be identified by

$$P_{false} - P_{lbist} = \{\ P \mid (P \in P_{false}) \wedge (\forall P\, i \in P_{lbist} : P\, i \neq P)\}$$

$\{P_{false} - P_{lbist}\}$ will also be included in the LBIST insertion constraint file.

**[00043]** To resolve the timing violations caused by control and observe point insertions, those points that are identified by $TP_{ctrl}\_vio$ and $TP_{obs}\_vio$ are removed from the test point insertion list. Thus, the new control and observe points to be inserted in the constrained LBIST insertion (the second time insertion) are $\{TP_{ctrl} - TP_{ctrl}\_vio\}$ and $\{TP_{obs} - TP_{obs}\_vio\}$, respectively.

**[00044]** For the inserted x-boundings that cause timing violations, these are first removed from the x-bounding list. The new x-bounding list thus becomes $\{TP_{xbnd} - TP_{xbnd}\_vio\}$, and timing violation problems are therefore resolved. The "X" values however, may propagate into scan flip-flops and later into the MISR. To block the "X" value propagations, substantially all flip-flops where an "X" value could be propagated into are identified. In an exemplary embodiment, this is performed by searching substantially all paths starting from the "X" source net. If the final device on the path is a flip-flop, its scan enable connection is then modified so that its capture is suppressed in LBIST mode. FIG. 7 is a circuit diagram illustrating this exemplary suppression approach for blocking "X" propagation. As shown, a suppression signal lbist_en 82 is received at the flipflop, marking the flip flop for suppression.

[00045] If the final device on the path is a memory input or an output, it is simply ignored since the "X" value will not be propagated to the MISR through a memory or an external output.

[00046] For timing violations in multi-cycle paths $\{P_{false} - P_{lbist}\}$, a similar method is used as in fixing x-bounding timing violations. First, the receiving flip-flops of all multi-cycle paths in $\{P_{false} - P_{lbist}\}$ are found and their capturing during LBIST mode is suppressed. This capture suppressing can avoid capturing unpredictable values due to the fact that the value propagation through these paths takes more than one clock cycle.

[00047] One advantage of the foregoing feature is that with the constrained LBIST insertion it is a practical guarantee that no timing violation for both functional mode and LBIST mode will exist in the LBIST inserted netlist of block 34 as shown in FIG. 2. The approach presented herein also minimizes the design efforts and possible schedule impact due to the introduction of LBIST. Since this approach is based on timing analysis to constrain the LBIST insertion, it practically guarantees that no iteration for re-synthesis occurs and only one iteration for timing analysis occurs.

[00048] It should be noted that the various features of the foregoing embodiments were discussed separately for clarity of description only and they can be incorporated in whole or in part into a single embodiment of the present invention having all or some these features. It should further be noted that the invention is not limited to ASICs but can readily used in conjunction with virtually any integrated circuit in which an LBIST testing approach is used.

[00049] While embodiments and applications of this invention have been shown and described, it would be apparent to those skilled in the art having the benefit of this disclosure that many more modifications than mentioned above are possible without departing from the inventive concepts herein. The invention, therefore, is not to be restricted except in the spirit of the appended claims.